A Refinement Approach for the Reuse of Privacy Risk Analysis Results

Sourya De, Daniel Le Métayer

A Refinement Approach for the Reuse of Privacy Risk Analysis Results

Sourya Joyee De and Daniel Le Métayer

Inria, Université de Lyon, France
Email: sourya-joyee.de@inria.fr, daniel.le-metayer@inria.fr

–
Published in: Proc. 5th Annual Privacy Forum (APF), pp. 52-83,
Springer, LNCS 10518, 2017
http://www.springer.com/cn/book/9783319672793

Abstract. The objective of this paper is to improve the cost effectiveness of privacy impact assessments through (1) a more systematic approach, (2) a better integration with privacy by design and (3) enhanced reusability. We present a three-tier process including a generic privacy risk analysis depending on the specifications of the system and two refinements based on the architecture and the deployment context respectively. We illustrate our approach with the design of a biometric access control system.

1 Introduction

With the adoption of the EU General Data Protection Regulation (GDPR) [18], conducting a data protection impact assessment will become mandatory for certain categories of personal data processing. A large body of literature has been devoted to data protection impact assessment and privacy impact assessment (*PIA*) [8–10], [19], [22], [31,32]. However, most of these papers focus on legal and organizational aspects and do not provide many details on the technical aspects of the impact assessment (*Privacy Risk Analysis* or *PRA*, here) [12], [14], which may be challenging and time consuming in practice. The objective of this paper is to fill this gap and to propose a methodology which can be applied to conduct a PRA in a systematic way, to use its results in the architecture selection process (following the *privacy by design* approach [4]) and to re-use its generic part for different products or deployment contexts. The aim of this work is therefore to improve the cost effectiveness of PIA through (1) a more systematic (and therefore repeatable) approach; (2) a better integration with privacy by design; and (3) enhanced reusability. Considering that some data controllers or technology providers may have to conduct many

PIAs for similar lines of products (or implementation variants), reusability can be a major factor in saving costs. Reflecting this need, Recital 92 of the GDPR [18] finds it "*reasonable and economic*" to carry out a single PIA for "*a common action or processing environment across an industry sector or segment or for a widely used horizontal activity*" and the recently published WP 29 guidelines [1] also encourage the development of sector-specific PIA frameworks.

The proposed analysis proceeds in three broad phases:

1. A generic privacy risk analysis phase which depends only on the specifications of the system and yields *generic harm trees*.
2. An architecture-based privacy risk analysis which takes into account the definitions of the possible architectures of the system and refines the generic harm trees into *architecture-specific harm trees*.
3. A context-based privacy risk analysis which takes into account the context of deployment of the system (e.g., a casino, an office cafeteria, a school) and further refines the architecture-specific harm trees into *context-specific harm trees*. Context-specific harm trees can be used to take decisions about the most suitable architectures.

To illustrate our approach, we consider the design of a biometric access control system. Such systems are now used commonly in many contexts such as border security controls, work premises, casinos, airports, chemical plants, hospitals, schools, etc. [5, 6]. However, the collection, storage and processing of biometric data raise complex privacy issues [23], [29], [40, 41], [2], [27], [35]. To deal with them, a wide array of dedicated techniques (such as secure sketches or fuzzy vaults) as well as adaptations of general privacy preserving techniques (such as homomorphic encryption or secure multi-party computation) have been proposed [3]. However, each technique is a building block solving specific privacy problems and suitable in specific contexts. In addition, a range of architectural options are generally possible to integrate these building blocks into a system. Therefore it would be beneficial to use the results of a privacy risk analysis to provide guidance to system designers and help them select a solution and justify it with respect to privacy risks.

In this paper, we choose the deployment of biometric access control systems in casinos as an illustration of the deployment context. The verification of the identities of casino customers is required by certain laws (to prevent access by minors or individuals on blacklists), which can justify the implementation of a biometric access control system to speed up the verification process [6].

We start with the definition of the terminology and some notions that are central to the paper in Section 2. In Section 3, we provide an overview of our three-phase approach before presenting each phase in sections 4, 5 and 6 respectively. We illustrate each phase with the biometric access control system introduced in Section 4. We discuss related works in Section 7 and conclude with avenues for further research in Section 8.

2 Preliminaries

In order to avoid any ambiguity about the terminology, we first introduce the key concepts used in the paper. The three main inputs of our PRA process are the *specification of the system*, the *architectures* and the *context*, which can be chatacterized as follows.

Definition 1. *The **specification of the system** is a high-level view of its functionalities and its interactions with its users (irrespective of any implementation).*

For example, the specification of a biometric access control system expresses that its goal is to grant access to authorized persons to a particular zone (e.g., office, casino, airport) based on their biometric identifiers. Biometric identifiers are collected during enrolment and stored as reference templates. During the access control phase, fresh biometric data is collected from the user, converted into a fresh template and compared with the stored template(s) using a pre-defined threshold. If the templates match, access control rules are used to grant or deny access. The specification does not contain any detail about the decomposition of the system into specific components, where each type of data is stored, where and how computations take place or who has control over the storage and processing units.

Definition 2. *An **architecture** includes the technical description of the components of the system (server, terminal, etc.), with their roles (storage, computation, etc.), the entities (system owners, users, etc.) controlling them and the data flows among them.*

A specification can generally be implemented by more than one architectures involving different components, performing different sets of functions, interacting in different ways and controlled by different entities.

Definition 3. *The **context** is defined as the environment (social, legal, economic, etc.) in which the system is deployed.*

For example, a biometric access control system may be implemented in a casino, an office cafeteria, an airport, to control access by employees, customers, travellers, etc. The context provides useful information about the possible misuses of the personal data and their likelihood.

Definition 4. *A **risk source** is any entity (individual or organization) that may process (legally or illegally) data belonging to a data subject and whose actions may directly or indirectly, intentionally or unintentionally lead to privacy harms.*

Examples of potential risk sources include cybercriminals, rogue system administrators and data controllers.

Definition 5. *A **feared event** is an event of the system that may lead to privacy harms.*

Examples of feared events include unauthorized access to personal data, use of personal data for unauthorized purposes and disclosure of personal data to unauthorized actors.

Definition 6. *A **privacy harm** is a negative impact of the use of the system on a data subject, or a group of data subjects (or society as a whole) as a result of a privacy breach.*

A wide variety of privacy harms can result from a feared event, including physical, mental, financial or reputation harms.

Definition 7. *A **harm tree** is a node-labeled tree describing the relationship among a privacy harm (root), feared events, risk sources and exploitations of personal data (leaves).*

The root node of a harm tree denotes a privacy harm. Leaf nodes represent the exploitation of data by the most likely risk source (for the root harm). Intermediate nodes represent the feared events caused by the risk sources. They can be seen as intermediate steps of potential privacy attacks. Children nodes are connected by an AND node if all of them are necessary to cause the parent node and by an OR node if any of them is sufficient. A harm tree can be associated with either an individual risk source or a group of risk sources, colluding or not, depending on the interactions needed to exploit the data. For conciseness, we do not discuss collusions in this paper but they can be dealt by the methodology.

The first objective of a risk analysis is to identify the privacy harms for a system in a given context and to assess the associated risks, generally

measured in terms of likelihood and severity. Several factors can influence privacy risks. The first one is the *exploitability* of personal data in the system, characterized by the set of resources (e.g., technical resources, access rights, background knowledge) needed by a risk source to exploit it. The dual notion is the *capacity* of a risk source, defined as the resources (e.g., technical resources, access rights, background knowledge) available to this risk source. Another main factor affecting the likelihood that a risk source may carry out an attack is its *motivation*, resulting from the balance between its incentives[1] and disincentives to cause a feared event or a harm. The exploitability of a data item depends only on the architecture, while the motivation of a risk source depends only on the context. The capacity of a risk source depends on both: access rights depend on the architecture, while background information and technical resources depend on the context.

We assume that the control over a component allows a risk source to get access to all its data (even though it is fully secure). Risk sources that do not have the control over a component can get access to its data only by attacking it, *persistently* or *transiently*. By transient exploitation of a component, we mean an exploitation for a short period of time or infrequent exploitations; by persistent exploitation we mean an exploitation of a component for a long period of time (e.g., for several days or months). Persistent exploitation is therefore more demanding than transient exploitation. To summarize, we consider four decreasing levels of power of a risk source over a component: (1) control over the component; (2) ability to perform persistent exploitation; (3) ability to perform transient exploitation and (4) inability to perform any exploitation.

3 General Approach

In this section, we provide an overview of our three-phase approach, leaving the details of each phase to the next sections. Figure 1 summarizes the inputs and outputs of each phase. In the remainder of the paper, the term "generic" refers to the types of privacy harms, risk sources and harm trees which depend only on the system specification[2].

Our approach is inspired by previous works on PRA [12–14], [16, 17], [36] while introducing three analysis levels to enhance reusability:

[1] Incentives should be taken in a general sense here, including lack of awareness in the case of unintentional breach.

[2] And are independent of the architecture and the context.

Phase 1 (*Generic PRA*) takes as inputs the specification and the generic components of the system and yields generic privacy harm trees. This phase has to be carried out only once for a given category of products, regardless of their architectures or deployment context. Its main steps are [12]:

- Definition of personal data involved;
- Definition of generic risk sources;
- Definition of generic feared events;
- Definition of generic privacy harms;
- Construction of generic harm trees.

Phase 2 (*Architecture-specific PRA*) takes as inputs the architectures to be analyzed and yields *architecture specific harm trees*. The main steps of Phase 2 (for each architecture) are:

- Definition of the exploitability values of personal data;
- Definition of relevant risk sources and their access rights;
- Refinement of generic harm trees to obtain harm trees specific to each architecture; the two refinement operations are the instantiation of generic components and the pruning of irrelevant subtrees.

Phase 3 (*Context-specific PRA*) takes as input the results of Phase 2 and the context of the deployment and yields a *context specific harm tree* for each architecture. It consists of:

- Definition of the background information available to the risk sources in the considered context (e.g., does the casino owner have enough information to identify a customer from his biometric data?).
- Definition of the technical resources available to the risk sources in this context (e.g., does an internal risk source have enough technical resources to get access to the access logs?)
- Definition of the motivation of each risk source for each feared event and harm (e.g., what are the incentives and disincentives for the employer to use biometric and access control data of his employees in order to track them?).
- Refinement of architecture-specific harm trees based on the results of the previous steps. The refinement operation in this phase is the pruning of irrelevant subtrees to remove unlikely or irrelevant scenarios (e.g., one may consider that, in a casino, the owner is unlikely to perform further surveillance of its customers).
- Computation of the likelihood of each relevant harm using context specific harm trees, the exploitability of personal data and the capacity and motivation of the risk sources.

We do not discuss the decision making step here, based on the result of the risk analysis, which typically involves opinions about acceptable risk levels and may take consider other factors such as costs and usability. The detailed description of the three phases and their illustration on a biometric access control system are presented in sections 4, 5 and 6 respectively.

The benefits of this incremental approach are two-fold:

1. **Reusability of intermediate results**: The results of Phase 1 (generic harm trees) can be reused for another implementation (or architecture) of the same type of system and the results of Phase 2 (architecture-specific harm trees) can be reused for the deployment of a product or system in a different context. This approach aligns with the WP 29 guidelines [1] which encourage the development of sector-specific PIAs and Recital 92 of the GDPR [18] that proposes the use of a single PIA to assess multiple operations that are similar in the risks they present.
2. **Selection of appropriate architecture in a privacy by design approach**: Because the results of Phase 1 do not depend on a specific architecture, they can be refined in different ways to consider different architectural options. Appropriate design decisions can be taken based on the results of the analysis of each option.

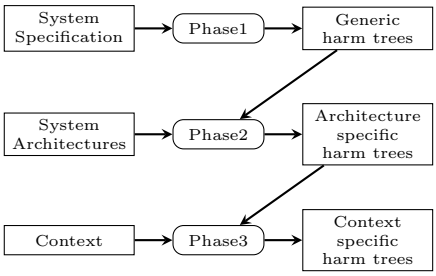

Fig. 1. Three phases of the selection process

4 Phase 1: Generic Privacy Risk Analysis

In this section, we present the first phase and illustrate it with our case study, the design of a biometric access control system. The following subsections describe successively the inputs of Phase 1 and the five steps introduced in Section 3.

4.1 Inputs: System specification and system components

The first step of a biometric access control system is the enrolment, involving the collection and storage of a biometric reference template br_i and identity ID_i for each user i of the system. As biometric data are sensitive, each reference template br_i is encrypted (into ebr_i) with a key (k_{br}) before being stored in a database ebr. Considering that some values are always stored with the identity of the user, we use the notation $\overline{x_i}$ (resp. $\overline{x}$) to denote the pair (x_i, ID_i) (resp. $\mathsf{list}(x_i, ID_i)$) for conciseness. The main authentication and access control steps are:

1. the input of fresh biometric raw data rd_i from the user i;
2. the conversion of rd_i into a fresh biometric template bs_i;
3. the comparison of bs_i with the enrolled template[3] br_i using a threshold thr;
4. if the templates match, the access control rules ac, are used to compute the decision dec_i (access grant or denial).

The system also manages an access trace (or access log) $\overline{at}$ consisting of the results of access control check dec_i and the associated time stamp ts_i along with the user's identity ID_i. Since access traces reveal information about users, they are usually stored as $\overline{eat}$, i.e., encrypted with a key k_{at}.

The components of a biometric access control system usually include a terminal T (C.1) to collect raw biometric data and a server S (C.2) to store information about users. In some cases, specific components such as a secure module M (C.3), a smart card C (C.4) or a second server $\mathsf{S'}$ (C.5) may also be used. The components on which the comparisons are performed and the encrypted biometric templates ebr_i are stored may vary depending on the architecture of the system. For example, the encrypted template may be stored on the server or on a smart card. Secure modules and smart cards are assumed to be tamper proof: only the actors controlling them can get access to their data.

4.2 Definition of generic data

The next step is the definition of the personal data processed by the system, which can be derived from its specification. Table 1 presents this list for the biometric access control system considered here. In a given architecture, each of these data is stored in one or more components, permanently or transiently. For example, the enrolled template ebr_i may

[3] The user's identity ID_i is used to fetch his enrolled template br_i.

be stored permanently in a database, and also transiently on the component performing the comparison with a fresh template. We assume that some data such as br_i and ebr_i are always associated with ID_i during enrolment (hence the use of $\overline{br_i}$ and $\overline{ebr_i}$ following our notation convention). So when a risk source has access to $\overline{br_i}$, it has also access to ID_i. For other data such as rd_i and bs_i, the identity ID_i may or may not be collected directly from the user during the access control phase. Therefore, we do not assume that they are always associated with ID_i. For example, in some scenarios, the user may be required to present a smart card containing his identity ID_i which is never transmitted to any of the components controlled by the owner (so that there is no trace of ID_i in these components although they may host rd_i and bs_i).

Code	**Data**
ID_i	Identity of user i
br_i	Biometric reference template of user i
ebr_i	Encrypted biometric reference template of user i
ebr	Encrypted database of biometric templates for all users
rd_i	Raw biometric data for user i
bs_i	Fresh biometric template derived from rd_i
dec_i	Decision (result of an access control check for user i)
ts_i	Time stamp associated with an access control of user i
$\overline{at}$	Access log of all users containing dec_i, ID_i and ts_i for all i
ac	Access control rules
k_{br}	Key used to encrypt and decrypt $\overline{ebr}$
k_{at}	Key used to encrypt and decrypt $\overline{at}$
$\overline{eat}$	Encrypted $\overline{at}$
thr	Threshold for comparing bs_i and $\overline{br_i}$

Table 1. Generic data

4.3 Definition of generic risk sources

We assume that each component may be controlled either by the system owner (data controller in the GDPR) or by a security operator acting as a sub-contractor of the owner. The precise set of components controlled by each actor depends on the architecture. For example, in some architectures, the security operator may control only the component performing the comparison. In other architectures, it may also control the component storing the reference templates. In addition to the system owner (A.1) and the security operator (A.2) who are internal risk sources, cybercriminals (A.3) and states (A.4) may act as external risk sources. In some cases, the system owner or the security operator may have business links with third parties (A.5) such as insurance providers or marketing companies, which may also become risk sources. In a real PRA, other risk sources such as employees of the owner and the operator should also be considered, but we do not discuss them here for space considerations.

4.4 Definition of generic feared events

Privacy harms result from the combination of one or more feared events. Generally speaking, we distinguish three types of feared events: the access to personal data, the use of personal data, and the disclosure of personal data. We consider two main types of personal data here, biometric data and access control results, which leads to the six generic feared events described in Table 2.

Code	Feared events
FE.1	Use of biometric data or data inferred from them for unauthorized purposes
FE.2	Use of result of biometric access control results and data inferred from them for unauthorized purposes
FE.3	Disclosure of biometric data to unauthorized actors
FE.4	Disclosure of results of biometric access controls to unauthorized actors
FE.5	Unauthorized access to biometric data
FE.6	Unauthorized access to results of biometric access controls

Table 2. Generic feared events for biometric access control systems

4.5 Definition of generic privacy harms

The possibility for a risk source to get access to access control results dec_i and access logs $\overline{at}$ makes the users of the system vulnerable to surveillance (H.1). Surveillance may also result from the misuse of biometric templates. It may be carried out by the system owner itself or the state (with different motivations). For example, an employer may try to find out how frequently a particular employee takes breaks based on the number of times he visits the cafeteria. Harms occur when surveillance takes place beyond the intended purpose of the access control system. Identity theft (H.2) is another important concern for biometric access control systems. It can be caused by wrongful access to biometric reference templates br_i, fresh biometric templates bs_i or even raw biometric data rd_i along with the user identity ID_i. Other harms are also possible (e.g., inference of sensitive attributes such as health data or genetic information, weight or body mass index [41], [35], [11]), but we do not discuss them here because of space limitations.

4.6 Construction of generic harm trees

Generic harm trees can be constructed for each of the harms discussed in Section 4.5 using the system components, risk sources and feared events identified in the previous subsections. In this section, we discuss only the generic harm tree for identity theft (H.2) (Figure 2). The interested reader can find the generic harm tree for surveillance (H.1) in [15]. Generic harm trees can be refined to specific components and risk sources when the details of the architectures and the context are available (Section 5 and Section 6). We use the notation $C.i$, $C.k$, etc. to denote generic components (which will be instantiated in the next phases) in the harm trees.

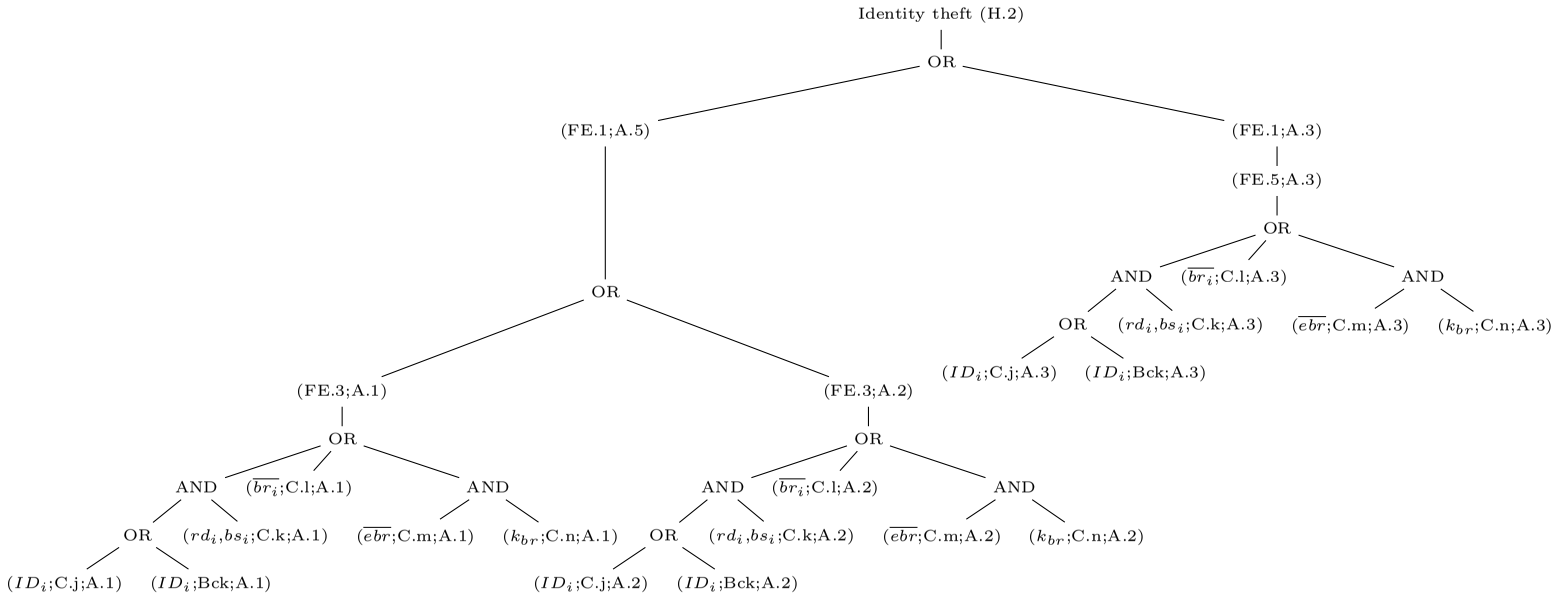

Fig. 2. Generic harm tree for identity theft (H.2) (Phase 1)

Figure 2 shows that the harm identity theft (H.2) can result from the use of biometric data for unauthorized purposes (FE.1). FE.1 itself can be caused by a cybercriminal (A.3) via unauthorized access to biometric data (FE.5) or by third parties (A.5) receiving biometric data (FE.3) from either the security operator (A.2) or the owner (A.1). FE.3 and FE.5 may be caused by the exploitation of different types of data in different components of the system. These exploitations of personal data are pictured by the leaves in the harm trees. Commas in the leaves are used as concise notations for disjunctions (e.g., rd_i, bs_i means rd_i OR bs_i).

Although theoretically possible, some combinations of risk sources and harms do not make sense in practice, irrespective of the details of the architecture or the context. For example, the system owner, the operator and the state are unlikely to perform identity theft. These combinations are left out of the generic harm trees. Therefore, Figure 2 does not have a branch where FE.1 is carried out by A.1 or A.2 or A.4.

ID_i may be obtained by a risk source either from a system component or as background information. These possibilities are differentiated by an OR subtree with two children in the harm trees. The abbreviation 'Bck' denotes background information. We assume that all other data can be obtained only from a system component (they are unlikely to be known as a background information by a risk source).

The generic harm tree only considers the most likely risk sources (with or without collusion) that may lead to a harm. When a harm is possible both via a single risk source or a collusion of risk sources, only the single risk source is represented (since it is less demanding and therefore more likely).

5 Phase 2: Architecture-specific Privacy Risk Analysis

Phase 2 takes as input the architecture(s) under consideration and specific system components (if any). Its goal is to refine the generic harm trees resulting from Phase 1 to obtain harm trees specific to each architecture. In this paper, we illustrate our approach with three architectures:

1. Arch.1, a simple architecture with an encrypted database,
2. Arch.2, an architecture with an encrypted database and a hardware security module and
3. Arch.3, an architecture relying on the match-on-card technology.

Due to space considerations, we describe only the treatment of Arch.2 in the main body of the paper. Phase 2 for Arch.1 and Arch.3 is described in Appendix A.

Figure 7 (Appendix A) shows the graphical representations of the biometric access control components used here. In the following subsections, the user and the enrolment site are not considered within the scope of the system. The issuer I is only involved in the enrolment phase. It is in charge of collecting and encrypting the enrolled biometric reference templates br_i along with user identities ID_i into $\overline{ebr_i}$ and storing them in the form of the database $\overline{ebr}$ in the server S. It has no role during the access control process and is included here for clarity only.

5.1 Description of Arch.2

In this architecture (Figure 3), a hardware security module M is used to compare the fresh template with the enrolled template, so that the clear template is never used in the terminal T. The module M is assumed to be managed by a security operator (A.2). The server S stores the database of encrypted reference templates $\overline{ebr}$ and the access control rules ac. A second server S' stores $\overline{ebr}$ (updated periodically from S to take new enrolments into account), ac (updated periodically from S) and $\overline{eat}$ (updated periodically by T).

When a user presents his identity ID_i and a fresh biometric rd_i to the terminal T, T computes bs_i, fetches ebr_i from S' and sends them to the module M. M decrypts $\overline{ebr_i}$ using the key k_{br}, compares br_i with bs_i (taking into account the threshold thr) and uses ac to compute dec_i which is returned to T and used to grant or deny access. The access log $\overline{at}$ is encrypted into $\overline{eat}$ by M and sent to T which stores it into S'.

The separate server S' controlled by the security operator (A.2) prevents the owner (A.1) from knowing the identity ID_i of a user requesting access. Moreover, the owner does not have access to clear biometric templates or results of access control checks. Therefore, the owner cannot carry out any surveillance or disclose biometric data to other risk sources. The owner's role is to devise access control rules, enroll users and inform the security operator A.2 about ac and $\overline{ebr}$ updates from time to time. The owner maintains a copy of $\overline{eat}$ for future reference (e.g., in case of a dispute with the user).

The keys k_{at} and k_{br}, the threshold thr and access control rules ac are stored in M. The decision dec_i is erased just after its use. Similarly, rd_i, bs_i, $\overline{ebr_i}$, $\overline{br_i}$, $\overline{eat}$, ts_i, $\overline{at}$, ID_i, and ts_i are deleted from the components (i.e., T and M) which use or generate them as soon as their use is over.

The system components in this architecture are the terminal T (C.1), the servers S (C.2) and S' (C.5) and the hardware security module M (C.3).

5.2 Risk sources for Arch.2

All risk sources have to be considered for Arch.2: the owner (A.1), the security operator (A.2), cybercriminals (A.3), the state (A.4) and third parties (A.5). We assume that the owner (A.1) controls only the server S while the security operator (A.2) controls the hardware security module M, the terminal T and the server S'. M is assumed to be secure and therefore cannot be (or is very unlikely to be) attacked.

5.3 Personal data and their exploitability values for Arch.2

Table 3 presents the personal data stored in each component with their exploitability values. Persistent exploitation is required to exploit the data stored on T because T stores these data only on a temporary basis. In contrast, transient exploitation is sufficient to exploit the data stored on S and S' which are used for long term storage. Since M is a secure component, the only possibility for a risk source to be able to exploit its data is to have control on it. Therefore, considering that the keys k_{br} and k_{at} are stored only on M, A.1, A.3 and A.4 cannot get access to them.

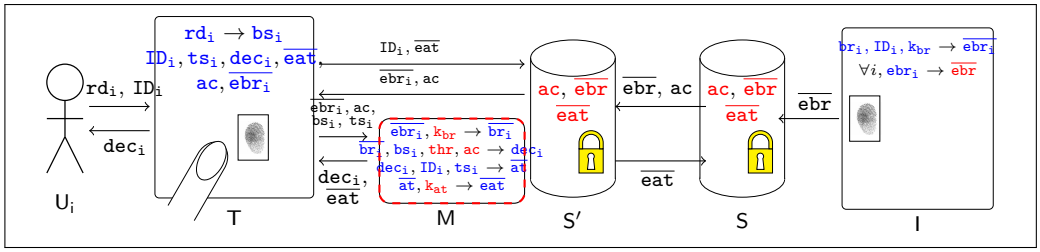

Fig. 3. Architecture Arch.2 : hardware security module (HSM)

In Figure 3 (also in figures 8 and 10 in Appendix A), all data elements in red colour inside a certain component (for example, ac in the server S) are stored persistently in the corresponding component whereas those in blue colour inside a component (for example, rd_i in the terminal T) are stored transiently in that component. We note that a data element that is stored persistently in one component can be stored transiently in another.

System component	Data	Exploitability
T	dec_i	Persistent exploit of T
T	ID_i	Persistent exploit of T
M	$\overline{at}$	Control of M
M	dec_i	Control of M
M	ID_i	Control of M
M	k_{br}	Control of M
M	$\overline{ebr_i}$	Control of M
M	$\overline{br_i}$	Control of M
S'	$\overline{ebr}$	Transient exploit of S'
S	$\overline{ebr}$	Transient exploit of S
T	rd_i, bs_i	Persistent exploit of T
M	rd_i, bs_i	Control of M
S'	$\overline{eat}$	Transient exploit of S'
M	k_{at}	Control of M

Table 3. Personal data in Arch.2 and their exploitability values

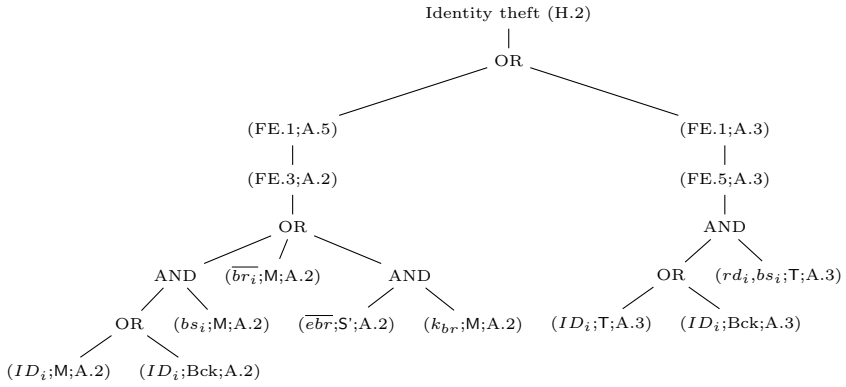

Fig. 4. Identity theft (H.2) harm tree for architecture Arch.2

5.4 Refinement of generic harm trees for Arch.2

Figure 14 in Appendix B shows how the generic harm tree for identity theft (H.2) (presented in Figure 2) can be pruned to derive the corresponding harm tree for Arch.2 (presented in Figure 4). In Arch.2, the owner of the system (A.1) has access only to S. Moreover, M is assumed to be a secure component. Therefore, no data element on any component other than S is accessible to A.1. So, A.1 can only access $\overline{ebr}$ (assuming that A.1 is unlikely to attack T for disclosing data to third parties (A.5)). However, to be able to exploit $\overline{ebr}$, the owner A.1 also needs to have access to k_{br} which is out of his reach since it is stored only in M. So, the branches in Figure 14 where A.1 needs access to $\overline{br_i}$ and k_{br} are pruned (marked with red cross). Similarly, a cybercriminal (A.3) cannot access the secure component M containing $\overline{br_i}$ and k_{br}. So the corresponding branches are pruned. Both rd_i and bs_i are accessible to the security operator A.2 as it controls both M and T. In the harm trees, for simplicity, we only show A.2's access to bs_i in M. The definition of the architecture helps to instantiate the generic components C_i, C_j, C_k, C_l, C_m and C_n (as shown in Figure 4).

6 Phase 3: Context-specific Privacy Risk Analysis

As described in Section 3, the objective of Phase 3 is to take into account all context specific factors. The harm trees specific to each architecture produced in Phase 2 (Section 5) are further pruned based on the deployment context. The likelihoods of the harms can then be computed based on these pruned trees, the exploitability values of the data and the capacities of the risk sources. The ultimate decision as to which architecture(s) is (are) more suitable can be taken based on these likelihoods and the severity of the harms. As discussed before, this decision may also generally involve other non-technical considerations.

6.1 Definition of the context

In this paper, we use casinos as an illustrative example of context. Casinos have to put in place strict checks to prevent the entry of individuals who are minors or blacklisted. To increase the efficiency of identity checks, some casinos want to implement biometric verification systems to control the visits of frequent customers. Users (frequent customers here) have to be initially enrolled by the owner (the casino here) to verify their identity. At this stage, the owner may also provide other relevant information

(such as the location of the casino[4]) that may later be useful to determine the capabilities and motivations of the risk sources. In the following subsections, we discuss the main contextual factors for our case study.

6.2 Definition of the background information available to risk sources

We assume that in this context, none of the risk sources is likely to possess the identity of the users as background information[5]. By availability of ID_i, we mean the availability of any information that can reveal ID_i.

6.3 Definition of the technical resources available to the risk sources

The system owner (A.1) and the security operator (A.2) are assumed to have technical resources for the transient exploitation of all components over which they do not have control. Third parties (A.5) also have technical resources for this transient exploitation. The state (A.4) and cybercriminals (A.3) are assumed to have the technical resources required for the persistent exploitation of any component.

The access rights of each risk source have already been specified in Phase 2. For a given architecture, the capabilities of each risk source can be derived by comparing the exploitability of the data and their technical resources and access rights. A risk source having control over a component has the highest capability (with respect to the data stored on this component) because it can exploit it irrespective of exploitability values. A risk source having technical resources for persistent exploitation also has high capability for data for which the exploitability value is persistent or transient and low otherwise. A risk source having technical resources for transient exploitation only has high capability for data with exploitability value equal to transient and low otherwise.

[4] For example, different locations correspond to different applicable laws (the motivation of a risk source may vary depending on the existence of data protection regulations and how strongly they are enforced), the strength (e.g., technical resources) or motivation of the local state to interfere [33], etc.

[5] This assumption should be valid at least for large scale attacks. However, one could argue that casinos may possess background information about certain frequent customers. Similarly, the state would be considered as having potentially a lot of background information but it is a more relevant risk source for surveillance than for identity theft. In any case, the assumptions made in this paper are for illustrative purposes only: different assumptions about background information could be made within the same framework.

6.4 Definition of the motivation of the risk sources

The motivations of the risk sources for the casino context are presented in Table 4. They depend on the feared events and sometimes also on specific harms. For example, the motivation of cybercriminals (A.3) to exploit biometric data for unauthorized purpose (FE.1) is high when the objective is identity theft (H.2) and medium for surveillance (H.1), since identity theft is a more lucrative scenario for cybercriminals, compared to surveillance. In contrast, the motivation of states (A.4) is high for surveillance to keep an eye on the citizens. The motivation for the casino owner to disclose data (FE.3, FE.4) or for unauthorized access to data (FE.5, FE.6) is only medium as such actions may have several incentives (such as monetary benefits from selling data) and many disincentives (such as bad reputation). Similarly, third parties (A.5) and security operators (A.2) may have several incentives and disincentives influencing their motivations.

Not all combinations of harms, feared events and risk sources are meaningful. For example, states are very unlikely to carry out identity theft against its own citizens. All unlikely combinations are marked with "×" under motivation in Table 4.

6.5 Final pruning of harm trees

The specific harm trees produced in Phase 2 can be further pruned depending on the contextual information (as described in sections 6.2, 6.3 and 6.4). For example, for the harm tree for Arch.2 in Figure 4, we observe that ID_i appears as background information in some of the branches. As discussed in Section 6.2, it is unlikely that any of the risk sources will possess ID_i as background information. Hence, the corresponding branches can be pruned. The pruned tree is shown Figure 5. Similarly, the harm trees for Arch.1 and Arch.3 can be pruned. For Arch.1, the pruned harm tree is shown in Figure 13 (Appendix B). For Arch.3, the pruning leads to an empty tree. The pruning is shown in Figure 16 in Appendix B.

Generally speaking, the context is of prime importance to distinguish relevant and irrelevant combinations of harms and risk sources. For example, casino owners are unlikely to track their customers beyond the purpose of the access control system. In contrast, an employer may be tempted to track his employees (e.g., to know how many breaks they take) beyond the purpose of the biometric access control system (e.g., to restrict the access of a cafeteria only to employees).

Risk Sources	Harms	Feared events	Motivation
Owner (A.1)	H.1	FE.3, FE.4, FE.5, FE.6	Medium
	H.2	FE.3, FE.5	Medium
	H.1	FE.1, FE.2	×
	H.2	FE.1	×
Security operator (A.2)	H.1	FE.1, FE.2	×
	H.2	FE.1	Low
	H.1	FE.3, FE.4, FE.5, FE.6	Medium
	H.2	FE.3, FE.5	Medium
Cybercriminal (A.3)	H.1	FE.1, FE.2, FE.3, FE.4, FE.5, FE.6	Medium
	H.2	FE.1, FE.3, FE.5	High
State (A.4)	H.1	FE.1, FE.2, FE.3, FE.4, FE.5, FE.6	High
	H.2	FE.1, FE.3, FE.5	×
Third party (A.5)	H.1	FE.1, FE.2, FE.3, FE.4, FE.5, FE.6	Medium
	H.2	FE.1, FE.3, FE.5	Medium

Table 4. Relevant risk sources and their motivations in the casino context

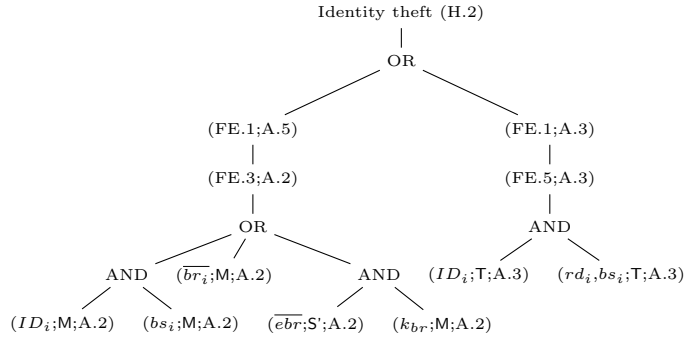

Fig. 5. Identity theft (H.2) final harm tree for architecture Arch.2

6.6 Computation of likelihoods based on harm trees

The computation of the likelihood of the harms based on the final harm trees can be carried out in two steps:

1. The first step is the assessment of the likelihood of the leaves of the harm trees (likelihood of exploitation of personal data) from the *motivation* and the *capability* of the relevant *risk sources*. This assessment is based on the motivations of the risk sources listed in Table 4 and the combination rules presented in Table 5.
2. The second step is the computation of the likelihood of each feared event and harm according to the following rules (applied bottom-up), where P_i is the likelihood of the ith child node:
   - R1. AND node with independent child nodes: $\prod_i P_i$.
   - R2. AND node with dependent child nodes[6]: $Min(P_i)$, i.e., minimum of the likelihoods of the child nodes.
   - R3. OR node with independent child nodes: $1 - \prod_i (1 - P_i)$.
   - R4. OR node with dependent child nodes[7]: $Min(1, \sum_i P_i)$.

For the computations of the second step, the symbolic likelihood values of Table 5 must be translated into numerical values. This transformation must be done by the privacy expert in collaboration with the owner and should be documented. In this paper, we use as an illustration the following correspondance for the likelihood values (p):

1. *Negligible (N)*: $p < 0.01\%$;
2. *Limited (L)*: $0.01\% \leq p < 0.1\%$;
3. *Intermediate (I)*: $0.1\% \leq p < 1\%$;
4. *Significant (S)*: $1\% \leq p < 10\%$;
5. *Maximum (M)*: $p \geq 10\%$.

Figure 6 depicts the computation of the likelihood for H.2 for Arch.2.

The likelihoods of the harms for the three architectures can be computed similarly (see Table 6). Needless to say, the analysis could lead to different results for different scenarios or different assumptions.

[6] In order to err on the safe side in terms of privacy protection, we consider dependent nodes such that one node may imply the other nodes.

[7] In order to err on the safe side in terms of privacy protection, we consider dependent nodes such that each node may exclude the other nodes.

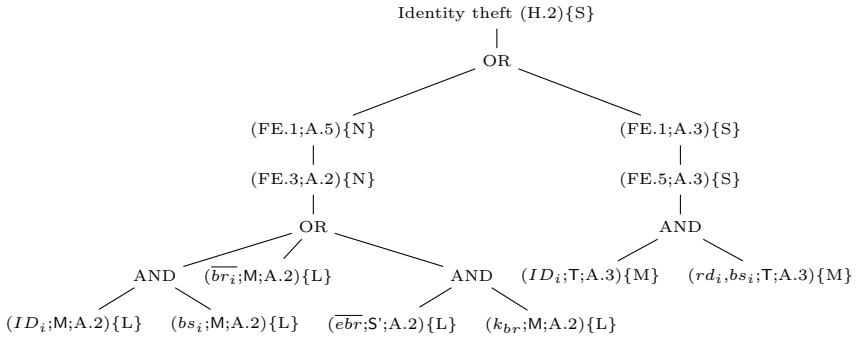

Fig. 6. Likelihood computation using the final pruned harm tree for identity theft (H.2) for architecture Arch.2 (after Phase 3)

6.7 Choice of architecture

The results of the previous sections can be used by the owner (with the help of the privacy expert and, ideally, after consultation of the stakeholders in the context of a PIA) to decide upon an acceptability threshold for each harm. Based on Table 6, and this threshold, he can select one or more acceptable architectures or decide to enhance them with further privacy protection measures. Let us assume that the system designer decides that the acceptability threshold for each of harm is "Limited". Then, none of the architectures considered here is acceptable. If the owner accepts "Significant" risks of state surveillance, then Arch.3 is the only acceptable architecture. The owner may be ready to accept a higher level of risk (for his customers) related to surveillance by the state and want to use Arch.2, as he does not want to manage the process related to the distribution and management of smart cards. Then, he has to decide (in collaboration with a privacy expert) upon additional counter-measures to reduce the risks. The harm tree in Figure 6 is a key source of information to make this decision. It shows that the target should be to better protect the terminal from cybercriminals.

If the storage of or an operation on a data element in a component seems to be a large contributor to the harm likelihood, one can think about replacing it by another component or reducing its role in the architecture. For example, the comparison between templates in the terminal (T) contributes more to the harm likelihood than doing it in the security module (M). So the harm tree helps to justify the roles of different components.

7 Related Works

In contrast with previous work on privacy by design, "privacy design strategies" or privacy engineering [26], [7], [24,25], [38], we do not propose a new design framework or process here, but a methodology to select an architecture among a range of options and to justify this choice with respect to a privacy risk analysis. Our work is therefore complementary to the above proposals and contributes to establish links between privacy risk analysis and privacy by design. The need to take into account the actual privacy risks or threats is mentioned in a number of papers [25], [34], [38] but, to our best knowledge, has not been explored in detail in previous works.

The notion of reusability is linked with the economics of problem solving [39]. It has been studied in the field of software engineering [28], [20] from both the economic and the technical viewpoints [30]. In this paper, we show how reusability can also be applied to privacy risk analysis. The framework presented in this paper builds on previous work on privacy risk analysis [12, 13] precisely to make reusability possible at several stages. Our methodology supports vertical reuse [37], i.e., the reuse of the generic harm trees resulting from Phase 1 for all architectures and the architecture-specific harm trees resulting from Phase 2 for all contexts. To our best knowledge, previous works on PRA or PIA [14], [17], [21], [31], [42] do not consider reusability.

Similar types of trees (sometimes called "threat trees" or "attack trees") have been used for PRA [12], [13], [16, 17], [36]. However, the focus of the work described here is not the risk analysis itself, but its adaptation and application to the architecture selection process. To this aim, we introduce generic harm trees and show how they can be successively refined.

Likelihood of exploitation	**Risk source capability**	**Motivation**
Negligible	Low	Low
Limited	High	
Negligible	Low	Medium
Significant	High	
Limited	Low	High
Maximum	High	

Table 5. Measurement rule for likelihood of exploitation

	Surveillance by the state (H.1,A.4)	**Identity theft (H.2)**
Encrypted Database (Arch.1)	Maximum	Maximum
HSM (Arch.2)	Maximum	Significant
Match-on-Card (Arch.3)	Significant	Negligible

Table 6. Comparison of the likelihoods of harms

8 Conclusion and Future Work

In this paper, we have presented a novel, incremental, approach to privacy risk analysis. We have also shown how the results of the analysis can be used by system designers to compare the privacy risks of different architectures and to choose the best option or find appropriate countermeasures.

We believe that establishing better links between privacy risk analysis and privacy by design is of prime importance in practice, especially in the context of the GDPR, which promotes both approaches. It is also important to improve the cost effectiveness of privacy risk analysis through the reuse and capitalization of results: in our framework, only the third phase has to be reconsidered in case of a change in the context; only the second and third phases for changes in the architectures; Phase 1 needs to be updated only when new types of privacy harms, feared events or risk sources emerge for a given system. This phase can be seen as a preliminary risk analysis valid for a whole line of products.

Another benefit of the three-phase process described here is a better clarity of the PRA process through a better separation of concerns.

One of the advantages of the order chosen here (considering first the specification, then the architectures and finally the context) is that the provider of a given solution (relying on a specific architecture) can build on the results of the second step to derive refined trees for different contexts (e.g. for different customers). In some situations however, it might be more efficient to consider the context before the architectures (e.g. to discard irrelevant harms). Space considerations prevent us from describing this option here but it is essentially a variant of the methodology described in this paper.

We have also not discussed certain features of the harm trees that can turn out to be useful in other contexts or for other systems or ar-

chitectures. For example, harm trees can include information about the possibility of collusion among risk sources. The motivations of the risk sources have to be properly defined when collusions are considered.

Last but not least, further types of risks (such as unavailability or loss of integrity) and considerations (such as usability and cost) have to be taken into account in practice. Any privacy risk that can be analyzed using harm trees can be dealt with by our methodology. As far as usability and costs are concerned, they have to be integrated in the decision process (which is not described in this paper as it can involve a variety of non-technical considerations).

Acknowledgements. This work has been partially funded by the French ANR-12-INSE-0013 project BIOPRIV and Inria Project Lab CAPPRIS.

A Description of Phase 2 for Arch.1 and Arch.3

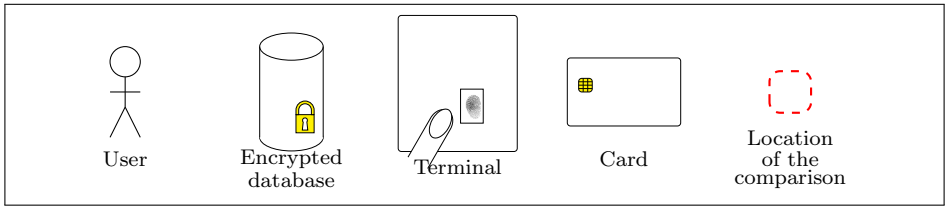

Fig. 7. Graphical representation of biometric access control systems

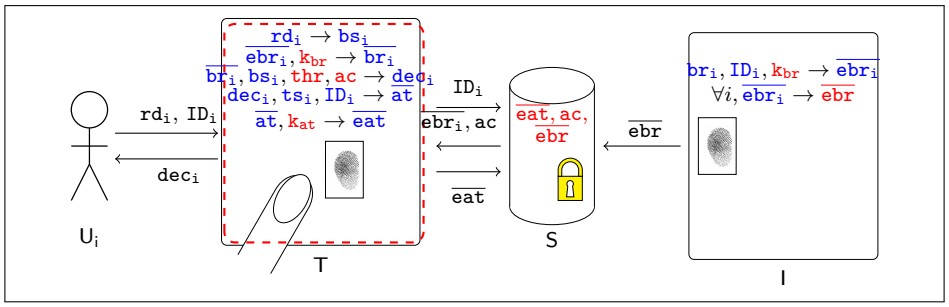

Fig. 8. Architecture Arch.1 : Encrypted database

A.1 Arch.1: Use of an Encrypted Database

Description of Arch.1 In the simple biometric access control architecture pictured in Figure 8, the server S stores the database of encrypted reference templates $\overline{ebr}$ and the access control rules ac. When the user presents his identity ID_i and fresh biometric rd_i to the terminal T, T fetches the encrypted reference template $\overline{ebr_i}$ from S, decrypts it using the key k_{br} and compares br_i with bs_i produced from rd_i by T (taking into account thr). The access control decision dec_i is used to allow or deny access. The access logs $\overline{at}$ of different users are encrypted into $\overline{eat}$ and sent back by the terminal T at regular intervals to be stored in the server S. The access log $\overline{at}$ is updated after each access control.

The keys[8] k_{at} and k_{br}, the threshold thr and access control rules ac are persistently stored in the terminal T[9]. In contrast, $\overline{at}$ is stored in T only for short time intervals. dec_i, rd_i, bs_i, $\overline{br_i}$, ts_i, $\overline{at}$, $\overline{eat}$, $\overline{ebr_i}$, ID_i are deleted from the terminal T as soon as their use is over[10].

The components in this architecture are therefore: the terminal T (C.1) and the server S (C.2).

Risk sources for Arch.1 Since the architecture does not include any security components, we assume that no security operator is involved. The risk sources are therefore: the owner (A.1), cybercriminals (A.3), the state (A.4) and third parties (A.5). The owner (A.1) controls both the server S and the terminal T.

Personal data for Arch.1 and their exploitability At this stage, the privacy analyst presents each data element stored in each system component and its exploitability (see Table 7). As explained in Section 2, by "transient exploitation" of a component we mean exploitation for a short period of time or infrequent exploitation, (e.g., once in several months), whereas "persistent exploitation" means the exploitation of a component for a long period of time (e.g., for several days or months). For example, dec_i provides the result of one access control for user i, whereas $\overline{at}$ provides the access log of all users for all previous days. So to know the access log of all users over t days, the risk source must either access all

[8] Keys are assumed to be protected by techniques which are not discussed here (e.g. obfuscation).

[9] Data elements that are stored persistently in a component are marked in red in Figure 8, Figure 3 and Figure 10.

[10] Data elements that are stored transiently in a component are marked in blue in Figure 8, Figure 3 and Figure 10.

System component	Data	Exploitability
T	dec_i	Persistent exploit of T
T	ID_i	Persistent exploit of T
T	$\overline{at}$	Transient exploit of T
S	$\overline{eat}$	Transient exploit of S
T	k_{at}	Transient exploit of T
T	k_{br}	Transient exploit of T
T	$\overline{ebr_i}$	Persistent exploit of T
T	$\overline{br_i}$	Persistent exploit of T
S	$\overline{ebr}$	Transient exploit of S
T	rd_i, bs_i	Persistent exploit of T

Table 7. Personal data in Arch.1 and their exploitability values

dec_i for all users for each of the t days (persistent exploitation) or access $\overline{at}$ at the end of the t days (transient exploitation).

Refinement of generic harm trees for Arch.1 In this phase, we consider the harm identity theft (H.2). Figure 9 shows the harm tree corresponding to this harm. Figure 12 in Appendix B shows how the generic harm tree (Figure 2) for identity theft is pruned to obtain the architecture specific harm tree in Figure 9. From Section A.1, we know that the risk sources for Arch.1 do not include A.2. Therefore, all branches of the generic harm tree for identity theft (H.2) that contain A.2 are pruned (pruned branches are marked by a red cross in Figure 12). The definition of the architecture also makes it possible to instantiate the generic components C_i, C_j, C_k, C_l, C_m and C_n.

A.2 Arch.3: Match-on-Card Technology

Description of Arch.3 Arch.2 is more protective than Arch.1 as the former uses a secure component M to perform the comparison between the fresh template and the reference template. In addition, it involves a security operator (A.2) for a better separation of responsibilities. However, in Arch.2, the fresh reference template bs_i is still available in T along

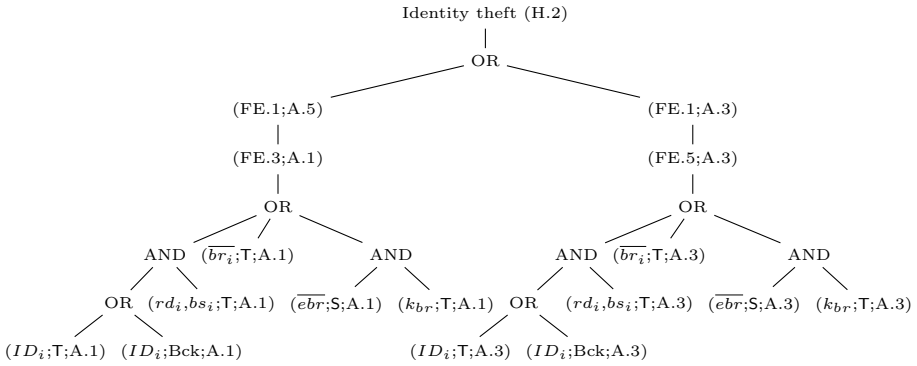

Fig. 9. Identity theft (H.2) harm tree for architecture Arch.1

with ID_i. Moreover, the clear template $\overline{br_i}$ can still be accessed by the security operator (A.2) who controls M. In fact, A.2 has access to a lot of personal data. One way to overcome these difficulties is to use the match-on-card technology. In Arch.3, pictured in Figure 10, each user possesses a smart card C that stores his identity ID_i along with his enrolled template br_i (i.e., it stores $\overline{br_i}$), the threshold thr and access control rules ac and performs the matching operation without disclosing ID_i or br_i to the terminal T. The owner does not store any database of reference templates.

The user inserts the card into the terminal T and submits the fresh biometric raw data rd_i. T derives a fresh template bs_i from rd_i and transfers it to C. C compares bs_i with br_i using the threshold thr and transfers the result of the access control dec_i to T. T informs the user about dec_i and sends it to the physical access control mechanism. The card C does not transfer any information apart from dec_i (not even the user identity ID_i) to T. C is assumed to be completely secure (e.g., it is tamper-resistant and personalized by a certified issuer during the enrolment phase). Both rd_i and bs_i as well as dec_i are deleted from T and C as soon as their uses are over. No access log $\overline{at}$ is recorded.

The system components in this architecture are: the terminal T (C.1) and the smart card C (C.4).

Risk sources for Arch.3 We assume that there is no security operator (A.2) in this architecture, since the security relies only on the smart cards possessed by the users. Therefore, the risk sources to be considered

include: the owner (A.1), cybercriminals (A.3), the state (A.4) and third parties (A.5). The owner (A.1) controls the terminal T.

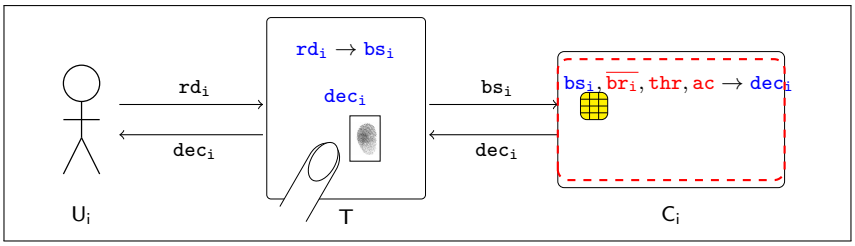

Fig. 10. Architecture Arch.3 : Match-On-Card technology

System component	Data	Exploitability
T	dec_i	Persistent exploit of T
T	rd_i, bs_i	Persistent exploit of T

Table 8. Personal data in Arch.3 and their exploitability values

Personal data and their exploitability for Arch.3 Table 8 presents each data item stored in each system component and the corresponding exploitability values for Arch.3. A risk source must have enough technical resources to exploit T persistently to get access to dec_i, rd_i or bs_i. However, in contrast with Arch.1 and Arch.2, ID_i is not stored in any component in Arch.3. Thus, in order to exploit dec_i or rd_i, bs_i, risk sources must have ID_i as background information. Since C is considered to be secure and belongs to the user, it does not appear in Table 8.

Refinement of generic harm trees for Arch.3 Figure 15 in Appendix B shows how the generic harm tree for identity theft (H.2) (presented in Figure 2) can be pruned to derive the corresponding harm tree for Arch.3 (presented in Figure 11). In Arch.3, ID_i, $\overline{br_i}$, $\overline{ebr_i}$ and k_{br} are not present at any moment in any of the components that the risk sources may access (i.e., terminal T). So all branches in the generic tree corresponding to these data elements are pruned. Also, the risk source A.2 is not a part

of Arch.3. So all branches concerning A.2 are pruned too. The definition of the architecture also makes it possible to instantiate the generic components C_i, C_j, C_k, C_l, C_m and C_n.

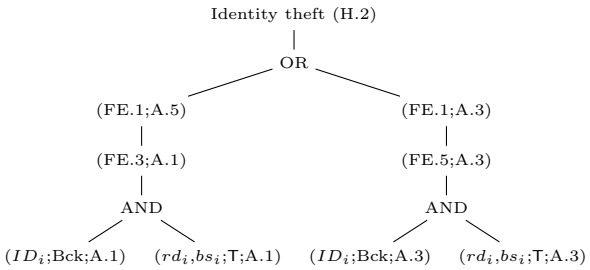

Fig. 11. Identity theft (H.2) harm tree for architecture Arch.3

B Pruning of harm trees and likelihood computation for identity theft (H.2)

In this appendix, we present the harm trees for identity theft, showing in detail how branches of the generic tree are pruned based on different conditions (related to the architecture and the context) discussed in the paper.

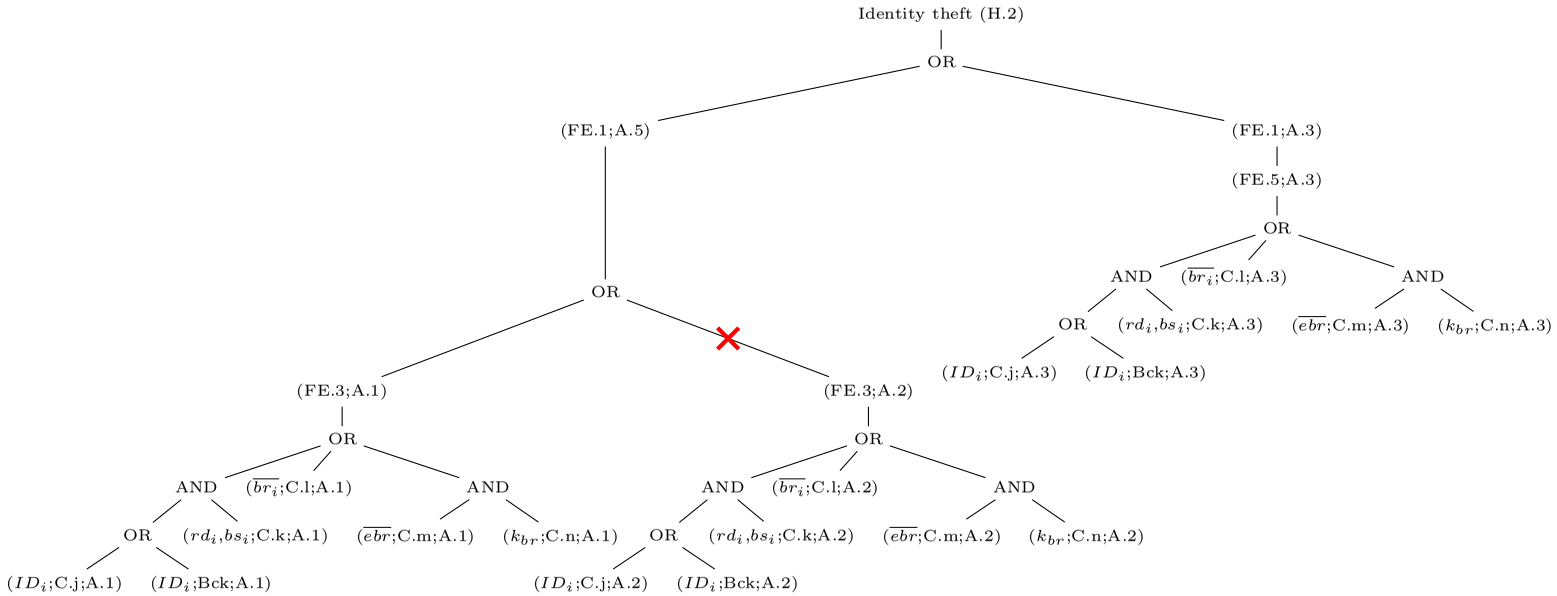

Fig. 12. Pruning of the generic harm tree for identity theft (H.2) to derive the harm tree for Arch.1 (Phase 2)

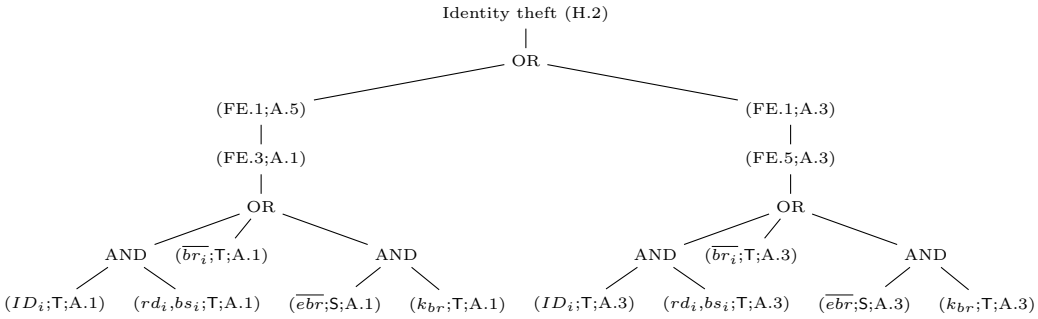

Fig. 13. Identity theft (H.2) final harm tree for architecture Arch.1

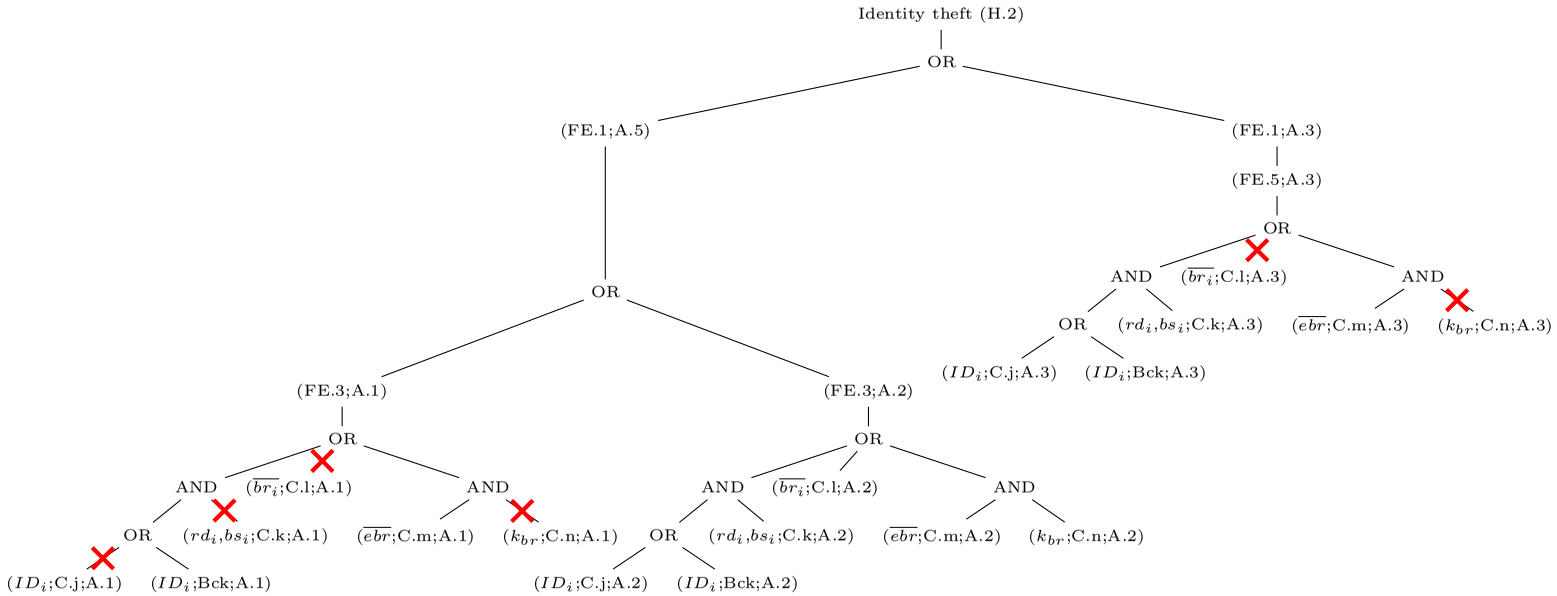

Fig. 14. Pruning of the generic harm tree for identity theft (H.2) to derive the harm tree for Arch.2 (Phase 2)

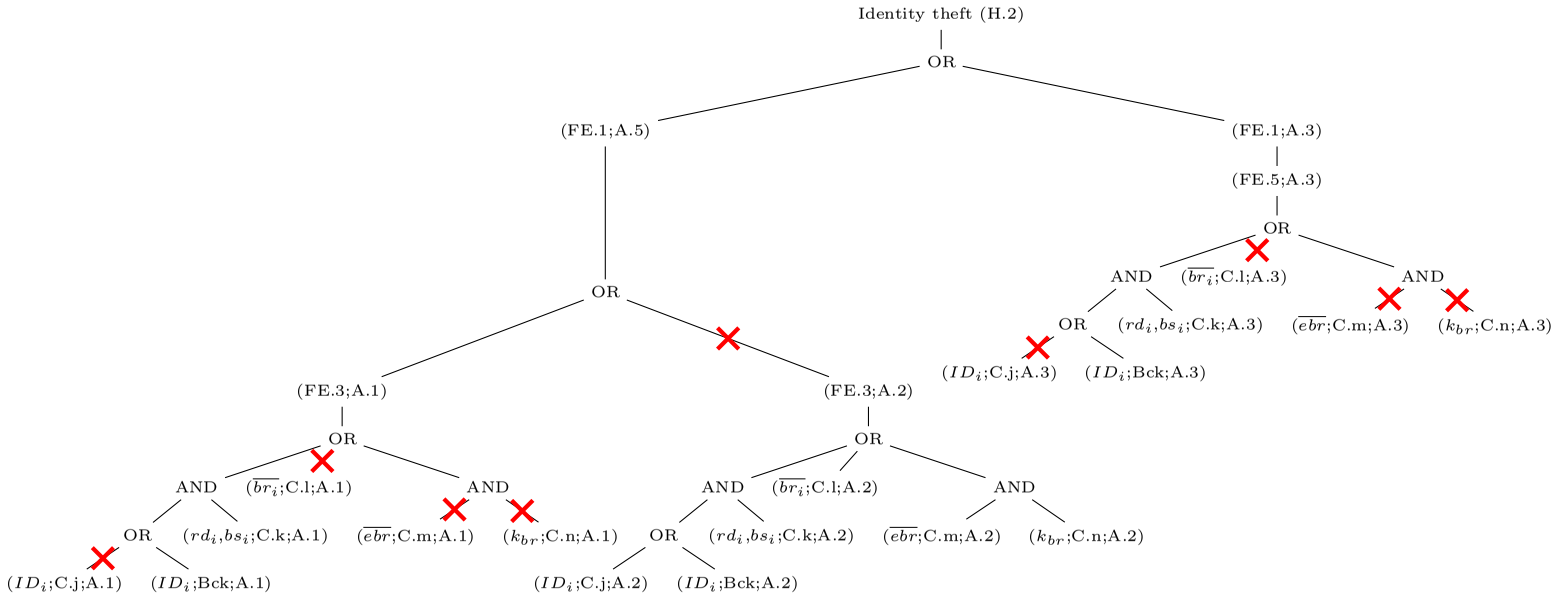

Fig. 15. Pruning of the generic harm tree for identity theft (H.2) to derive the harm tree specific to Arch.3 (Phase 2)

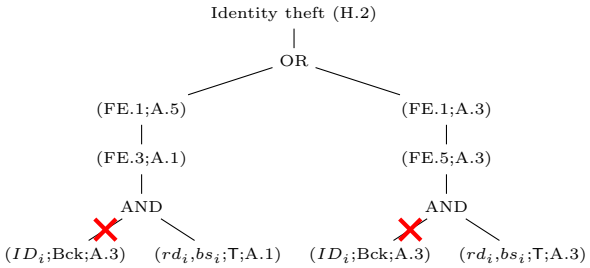

Fig. 16. Final pruning of the harm tree for identity theft (H.2) for architecture Arch.3 (Phase 3)

References

1. Article 29 Data Protection Working Party: Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is "likely to result in a high risk" for the purposes of Regulation 2016/679 (2017)
2. BBC Technology: Millions of Fingerprints Stolen in US Government Hack (2015)
3. Bringer, J., Chabanne, H., Métayer, D.L., Lescuyer, R.: Privacy by design in practice: Reasoning about privacy properties of biometric system architectures. In: FM 2015: Formal Methods - 20th International Symposium, Oslo, Norway, June 24-26, 2015, Proceedings. pp. 90–107 (2015)
4. Cavoukian, A.: Privacy by Design: The 7 Foundational Principles Implementation and Mapping of Fair Information Practices. Office of the Information and Privacy Commissioner, Ontario, Canada Standards (2010)
5. Cavoukian, A., Chibba, M., Stoianov, A.: Advances in Biometric Encryption: Taking Privacy by Design From Academic Research to Deployment. Review of Policy Research 29(1), 37–61 (2012)
6. Cavoukian, A., Stoianov, A.: Privacy by Design Solutions for Biometric One-to-Many Identification Systems (2014)
7. Colesky, M., Hoepman, J., Hillen, C.: A Critical Analysis of Privacy Design Strategies. In: 2016 IEEE Security and Privacy Workshops, SP Workshops 2016, San Jose, CA, USA, May 22-26, 2016. pp. 33–40 (2016)
8. Commission Nationale de l'Informatique et des Libertes (CNIL): Methodology for Privacy Risk Management – How to Implement the Data Protection Act (2012)
9. Commission Nationale de l'Informatique et des Libertes (CNIL): Privacy Impact Assessment (PIA) Methodology (How to Carry Out a PIA) (2015)
10. Commission Nationale de l'Informatique et des Libertes (CNIL): Privacy Impact Assessment (PIA) Tools (templates and knowledge bases) (2015)
11. Dantcheva, A., Elia, P., Ross, A.: What Else Does Your Biometric Data Reveal? A Survey on Soft Biometrics (2015)
12. De, S.J., Le Métayer, D.: PRIAM: A Privacy Risk Analysis Methodology. In: 11th International Workshop on Data Privacy Management (DPM). IEEE (2016)
13. De, S.J., Le Métayer, D.: Privacy Harm Analysis: A Case Study on Smart Grids. In: International Workshop on Privacy Engineering (IWPE). IEEE (2016)

14. De, S.J., Le Métayer, D.: Privacy Risk Analysis. In: Synthesis Series. Morgan & Claypool Publishers (2016)
15. De, S.J., Le Métayer, D.: A Risk-based Approach to Privacy by Design (Extended Version). No. RR-9001 (December, 2016)
16. De, S.J., Le Métayer, D.: PRIAM: A Privacy Risk Analysis Methodology. INRIA Research Report (RR-8876) (July, 2016)
17. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A Privacy Threat Analysis Framework: Supporting the Elicitation and Fulfilment of Privacy Requirements. Requirements Engineering 16(1), 3–32 (2011)
18. European Commission: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016)
19. Expert Group 2 of Smart Grid Task Force: Data Protection Impact Assessment Template for Smart Grid and Smart Metering Systems (2014)
20. Frakes, W.B., Kang, K.: Software reuse research: Status and future. IEEE transactions on Software Engineering 31(7), 529–536 (2005)
21. Friginal, J., Guiochet, J., Killijian, M.O.: A privacy risk assessment methodology for location-based systems. http://homepages.laas.fr/guiochet/telecharge/MOBIQUITOUS2013.pdf, accessed: 2016-07-13
22. Garcia, M., Lefkovitz, N., Lightman, S.: Privacy Risk Management for Federal Information Systems (NISTIR 8062 (Draft)). National Institute of Standards and Technology (2015)
23. Gartland, C.: Biometrics Are a Grave Threat to Privacy (2016), the New York Times
24. Gürses, S., Troncoso, C., Diaz, C.: Engineering Privacy by Design. Computers, Privacy & Data Protection 14(3) (2011)
25. Gürses, S., Troncoso, C., Diaz, C.: Engineering Privacy by Design Reloaded (2015)
26. Hoepman, J.H.: Privacy Design Strategies. In: IFIP International Information Security Conference. pp. 446–459. Springer (2014)
27. Kobie, N.: Surveillance State: Fingerprinting Pupils Raises Safety and Privacy Concerns (2016), the Guardian
28. Mcilroy, M.: Mass produced software components (1969)
29. Miglani, S., Kumar, M.: India's Billion-member Biometric Database Raises Privacy Fears (2016), reuters
30. Mili, A., Chmiel, S.F., Gottumukkala, R., Zhang, L.: An integrated cost model for software reuse. In: Software Engineering, 2000. Proceedings of the 2000 International Conference on. pp. 157–166. IEEE (2000)
31. Oetzel, M.C., Spiekermann, S.: A Systematic Methodology for Privacy Impact Assessments: A Design Science Approach. European Journal of Information Systems 23(2), 126–150 (2014)
32. Oetzel, M.C., Spiekermann, S., Grüning, I., Kelter, H., Mull, S.: Privacy Impact Assessment Guideline for RFID Applications (2011)
33. Oppenheim, C.: Big Brother Spying is Reaching Scary Levels. http://edition.cnn.com/2013/12/10/opinion/oppenheim-privacy-reform/ (2013)
34. Pearson, S., Benameur, A.: A Decision Support System for Design for Privacy. In: IFIP PrimeLife International Summer School on Privacy and Identity Management for Life. pp. 283–296. Springer (2010)
35. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric Recognition: Security and Privacy Concerns. IEEE Security & Privacy (2), 33–42 (2003)

36. Nunez del Prado Cortez, M., Friginal, J.: Geo-Location Inference Attacks: From Modelling to Privacy Risk Assessment. In: Tenth European Dependable Computing Conference (EDCC). pp. 222–225. IEEE (2014)
37. Prieto-Díaz, R.: Status Report: Software Reusability. IEEE software 10(3), 61–66 (1993)
38. Spiekermann, S., Cranor, L.F.: Engineering Privacy. IEEE Transactions on software engineering 35(1), 67–82 (2009)
39. Standish, T.A.: An Essay on Software Reuse. IEEE Transactions on Software Engineering (5), 494–497 (1984)
40. Tillman, G.: Opinion: Stolen Fingers: The Case Against Biometric Identity Theft Protection (2009), computer World
41. Woodward, J.D.: Biometrics: Privacy's Foe or Privacy's Friend? Proceedings of the IEEE 85(9), 1480–1492 (1997)
42. Wright, D., De Hert, P.: Privacy Impact Assessment. Springer (2012)